The claims

The status of the claims is as follows:

1. (Previously presented) An integrated circuit processor comprising:

a first instruction buffer corresponding to a primary thread;

a second instruction buffer corresponding to a backup thread;

a thread switch mechanism that detects when the primary thread stalls, and in response thereto, swaps instructions stored in the first instruction buffer with instructions stored in the second instruction buffer.

2. (Original) The integrated circuit processor of claim 1 wherein execution of the backup thread occurs after the swap by executing at least one instruction in the first instruction buffer.

3. (Previously presented) The integrated circuit processor of claim 1 further comprising:

a third instruction buffer corresponding to a second primary thread;

a fourth instruction buffer corresponding to a second backup thread;

wherein the thread swap mechanism further detects when the second primary thread stalls, and in response thereto, swaps instructions stored in the third instruction buffer with instructions stored in the fourth instruction buffer.

4. (Original) The integrated circuit processor of claim 3 wherein the first and second primary threads simultaneously issue instructions for execution.

5. (Previously presented) An integrated circuit processor comprising:

a first primary instruction buffer corresponding to a first primary thread;

a second primary instruction buffer corresponding to a second primary thread;

wherein the first and second primary threads simultaneously issue instructions for execution;

a first backup instruction buffer;

a second backup instruction buffer;

a thread switch mechanism that detects when one of the first and second threads stalls, and in response thereto, swaps instructions stored in one of the first and second primary instruction buffers corresponding to the stalled thread with instructions stored in one of the first and second backup instruction buffers.

6. (Original) The integrated circuit processor of claim 5 wherein the thread switch mechanism:

(1) detects when the first primary thread stalls, and in response thereto, swaps the first primary instruction buffer with the first backup instruction buffer; and

(2) detects when the second thread stalls, and in response thereto, swaps the second primary instruction buffer with the second backup instruction buffer.

7. (Previously presented) The integrated circuit processor of claim 5 wherein the first and second backup instruction buffers are part of a pool of backup instruction buffers, wherein instructions in any backup instruction buffer in the pool may be swapped with instructions in the first primary instruction buffer, and wherein instructions in any backup instruction buffer in the pool may be swapped with instructions in the second primary instruction buffer.

8. (Previously presented) An integrated circuit processor comprising:

a first primary instruction buffer corresponding to a first primary thread;

a second primary instruction buffer corresponding to a second primary thread;

wherein the first and second primary threads simultaneously issue instructions for execution;

a first backup instruction buffer;

a second backup instruction buffer;

a thread switch mechanism that detects when the first thread stalls, and in response thereto, swaps instructions stored in the first primary instruction buffer with instructions stored in the first backup instruction buffer, and begins issuing from the first primary instruction buffer, and that detects when the second thread stalls, and in response thereto, swaps instructions stored in the second primary instruction buffer with instructions stored in the second backup instruction buffer, and begins issuing from the second primary instruction buffer.

9. (Previously presented) A method for switching between a first thread of execution and a second thread of execution in a multithreaded processor, the method comprising the steps of:

(A) providing a first instruction buffer corresponding to the first thread;

(B) providing a second instruction buffer corresponding to the second thread;

(C) swapping instructions stored in the first instruction buffer with instructions stored in the second instruction buffer.

10. (Original) The method of claim 9 wherein step (C) is performed when switching between the first thread and the second thread is required.

11. (Original) The method of claim 9 wherein step (C) is performed when the first thread stalls.

12. (Original) The method of claim 9 wherein step (C) is performed when the second thread stalls.

13. (Previously presented) The method of claim 9 further comprising the step of executing the second thread after the swapping of instructions in step (C) by executing at least one instruction in the first instruction buffer.

14. (Previously presented) The method of claim 9 further comprising the steps of:

(D) providing a third instruction buffer corresponding to a third thread;

(E) providing a fourth instruction buffer corresponding to a fourth thread; and

(F) swapping instructions stored in the third instruction buffer with instructions stored in the fourth instruction buffer.

15. (Original) The method of claim 14 wherein step (F) is performed when the third thread stalls.

16. (Original) The method of claim 14 wherein step (F) is performed when the fourth thread stalls.

17. (Original) The method of claim 14 wherein the first and third threads simultaneously issue instructions for execution.

18. (Previously presented) A method for switching between first and second threads of execution in a multithreaded processor, the method comprising the steps of:

(A) providing a first primary instruction buffer corresponding to the first thread;

(B) providing a second primary instruction buffer corresponding to the second thread;

(C) providing a first backup instruction buffer corresponding to a first backup thread;

(D) providing a second backup instruction buffer corresponding to a second backup thread;

(E) simultaneously issuing instructions from the first primary instruction buffer and from the second primary instruction buffer; and

(F) detecting when one of the first and second primary threads stalls, and in response thereto, swapping instructions stored in one of the first and second primary instruction buffers corresponding to the stalled thread with instructions stored in one of the first and second backup instruction buffers.

19. (Previously presented) The method of claim 18 wherein step (E) comprises the steps of:

(1) detecting when the first primary thread stalls, and in response thereto, swapping information stored in the first primary instruction buffer with information stored in the first backup instruction buffer; and

(2) detecting when the second thread stalls, and in response thereto, swapping instructions stored in the second primary instruction buffer with instructions stored in the second backup instruction buffer.

20. (Previously presented) The method of claim 18 wherein the first and second backup instruction buffers are part of a pool of backup instruction buffers, wherein instructions in any backup instruction buffer in the pool may be swapped with instructions in the first primary instruction buffer, and wherein instructions in any backup instruction buffer in the pool may be swapped with instructions in the second primary instruction buffer.

21. (Previously presented) A method for switching between threads of execution in a multithreaded processor, the method comprising the steps of:

(A) providing a first primary instruction buffer corresponding to the first thread;

(B) providing a second primary instruction buffer corresponding to the second thread;

(C) providing a first backup instruction buffer corresponding to a first backup thread;

(D) providing a second backup instruction buffer corresponding to a second backup thread;

(E) simultaneously issuing instructions from the first primary instruction buffer and from the second primary instruction buffer; and

(F) detecting when the first threads stalls, and in response thereto, swapping instructions stored in the first primary instruction buffer with instructions stored in the first backup instruction buffer, and issuing instructions from the first primary instruction buffer.

22. (Previously presented) The method of claim 21 further comprising the step of

(G) detecting when the second thread stalls, and in response thereto, swapping instructions stored in the second primary instruction buffer with instructions stored in the second backup instruction buffer, and issuing from the second primary instruction buffer.

STATUS OF THE CLAIMS

Claims 1-22 were originally filed in this patent application. In response to the office action dated 03/29/2006, a Request for Reconsideration was filed on 06/29/2006 that included a declaration under 37 C.F.R. 1.132 to remove the Luick reference as prior art. In response to the office action dated 09/14/2006, an RCE and amendment was filed on 09/27/2006. In the pending office action, claims 1-2 and 9-13 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,348,671 to Doing *et al.* (hereinafter "Doing") in view of U.S. Patent No. 5,933,627 to Parady and claims 3-6, 8, 14-19, 21 and 22 were rejected under 35 U.S.C. §103(a) as being unpatentable over Doing in view of Parady and further in view of U.S. Patent Application Publication No. 2003/0135711 to Shoemaker *et al.* (hereinafter "Shoemaker"). Claims 7 and 20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Doing in view of Parady, Shoemaker and U.S. Patent No. 6,314,511 to Levy *et al.* (hereinafter "Levy"). Claims 7 and 20 were also rejected under 35 U.S.C. §103(a) as being unpatentable over Doing in view Parady and Shoemaker. No claim was allowed. No claims have been amended herein. Claims 1-22 are currently pending.